Discrete Fourier Transform

Definition

- The discrete Fourier transform (DFT) $X[k]$ of a length-N time-domain sequence $x[n]$ is defined by

$$\boxed{X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}, \ 0 \le k \le N-1}$$

- Note: The DFT $X[k]$ is also a length-N sequence in the integer variable k

---

Discrete Fourier Transform

- Sometimes, the length-N DFT sequence is referred to as the N-point DFT
- Note: Each sample of the DFT, in general, is a complex number
- As the DFT of a finite-length sequence with finite sample values is computed using a finite sum, the DFT always exists

---

Discrete Fourier Transform

Example – Determine the 4-point DFT $\{G[k]\}$, $0 \le k \le 3$ of the length-4 sequence

$$\{g[n]\} = \{2, \ -3, \ 1, \ 4\}, \ 0 \le n \le 3$$

- Now,

$$G[k] = \sum_{n=0}^{3} g[n]e^{-j2\pi kn/4} = \sum_{n=0}^{3} g[n]e^{-j\pi kn/2},$$

$$0 \le k \le 3$$

Discrete Fourier Transform

- Thus,

$$G[0] = g[0] + g[1] + g[2] + g[3] = 2 - 3 + 1 + 4 = 4$$

$$G[1] = g[0] + g[1]e^{-j\pi/2} + g[2]e^{-j\pi} + g[3]e^{-j3\pi/2}$$

$$= 2 + j3 - 1 + j4 = 1 + j7$$

$$G[2] = g[0] + g[1]e^{-j\pi} + g[2]e^{-j2\pi} + g[3]e^{-j3\pi}$$

$$= 2 + 3 + 1 - 4 = 2$$

$$G[3] = g[0] + g[1]e^{-j3\pi/2} + g[2]e^{-j3\pi} + g[3]e^{-j9\pi/2}$$

$$= 2 - j3 - 1 - j4 = 1 - j7$$

Discrete Fourier Transform

- Hence,

$$\{G[k]\} = \{4, \ 1 + j7, \ 2, \ 1 - j7\}, \ 0 \le k \le 3$$

- We shall show later that the samples of the N-point DFT sequence $X[k]$ are given by the samples of the DTFT $X(e^{j\omega})$ at N equally-spaced points on the angular frequency ω-axis from 0 to 2π

Discrete Fourier Transform

- Hence, the integer variable k is in the frequency domain
- For a given N, the spacing $2\pi/N$ between two consecutive DFT samples is called the resolution of the DFT
- In some applications, the length of the DFT sequence can be larger than that of the parent time-domain sequence providing higher resolution

Discrete Fourier Transform

- To compute an L-point DFT $X[k]$ of a length-N sequence $x[n]$ with $L > N$, we add $L - N$ zero-valued samples at the end of the sequence $x[n]$ resulting in a length-L sequence $x_e[n]$ given by

$$\boxed{x_e[n] = \begin{cases} x[n], & 0 \le n \le N-1 \\ 0, & N \le n \le L-1 \end{cases}}$$

Discrete Fourier Transform

- The process of adding zero-valued samples to a sequence is called appending with zeros
- The length-L DFT $X_e[k]$ is then given by

$$\boxed{\begin{aligned} X_e[k] &= \sum_{n=0}^{L-1} x_e[n] e^{-j2\pi kn/L} \\ &= \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/L}, \ 0 \le k \le L-1 \end{aligned}}$$

Discrete Fourier Transform

- Note: The DFT being a sequence, most of the basic operations on time-domain sequences such as addition, subtraction, amplitude scaling, modulation, and division described earlier can also be applied to DFTs of same length and defined for the same frequency ranges

Discrete Fourier Transform

- The inverse discrete Fourier transform (IDFT) of the N-point DFT $X[k]$ is a length-N sequence $x[n]$ given by

$$\boxed{x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi kn/N}, \quad 0 \le n \le N-1}$$

- An often-used short-hand notation for the complex number $e^{-j2\pi/N}$ is W_N, that is,

$$\boxed{W_N = e^{-j2\pi/N}}$$

Discrete Fourier Transform

- Using this notation the modified expressions for the DFT and the IDFT are

$$\boxed{\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n] W_N^{kn}, \ 0 \le k \le N-1 \\ x[n] &= \sum_{k=0}^{N-1} X[k] W_N^{-kn}, \ 0 \le n \le N-1 \end{aligned}}$$

Discrete Fourier Transform

- The DFT-IDFT pair are often written in compact form as

$$\boxed{x[n] \overset{\text{DFT}}{\leftrightarrow} X[k]}$$

- The complex exponential sequence $W_N^n = e^{-j2\pi n/N}$ is a periodic sequence of n with a fundamental period N as

$$W_N^{rN+n} = W_N^{rN} W_N^n = W_N^n$$

Discrete Fourier Transform

- An important identity involving this sequence is

$$\boxed{\frac{1}{N}\sum_{n=0}^{N-1} W_N^{(k-\ell)n} = \begin{cases}1, & \text{for } k=\ell+rN\\ 0, & \text{for } k\neq \ell\end{cases}}$$

Example – Consider the length-N sequence

$$y[n]=\alpha^n,\ 0\le n\le N-1$$

- Its N-point DFT is given by

Discrete Fourier Transform

$$Y[k]=\sum_{n=0}^{N-1}\alpha^n W_N^{kn}=\sum_{n=0}^{N-1}\left(\alpha W_N^k\right)^n$$

$$=\frac{1-\alpha^N W_N^{kN}}{1-\alpha W_N^k}=\frac{1-\alpha^N}{1-\alpha W_N^k},0\le k\le N-1$$

- In compact form

$$\boxed{\alpha^n \overset{\text{DFT}}{\leftrightarrow} \frac{1-\alpha^N}{1-\alpha W_N^k},\ \alpha\neq 1}$$

Discrete Fourier Transform

Example – Consider the N-point DFT

$$X[k]=\delta[k]\ ,\ 0\le k\le N-1$$

- Its length-N IDFT is given by

$$x[n]=\frac{1}{N}\sum_{k=0}^{N-1}\delta[k]W_N^{-kn}=\frac{1}{N}\ ,\ 0\le n\le N-1$$

- In compact form

$$\boxed{\frac{1}{N}\overset{\text{DFT}}{\leftrightarrow}\delta[k]}$$

DFT Computation Using MATLAB

- MATLAB functions for the computation of the DFT and the IDFT are based on fast Fourier transform (FFT) algorithms
- Later in the course, we shall describe the basic idea behind one such algorithm
- The function `fft(x)` generates the DFT sequence of same length as the time-domain sequence `x`

DFT Computation Using MATLAB

- The function `ifft(X)` generates the IDFT sequence of same length as the DFT sequence `X`
- The function `fft(x,L)` generates the L-point DFT sequence of the length-N time-domain sequence `x` where `L > N`

DFT Computation Using MATLAB

- The function `ifft(X,L)` generates the length-L time-domain sequence of the N-point DFT sequence `X` where `L > N`

Example – We determine the 4-point DFT of

$$\{g[n]\}=\{2,\ -3,\ 1,\ 4\},\ 0\le n\le 3$$

- Code fragments used are given in the next slide

DFT Computation Using MATLAB

```
g = [2  -3  1  4];
G = fft(g);
```

which yield

```
G =
4.0000  1.0000+7.000i  2.000
1.000-7.000i
```

DFT Computation Using MATLAB

Example – We determine the length-4 IDFT of $\{G[k]\} = \{4,\ 1+j7,\ 2,\ 1-j7\}$, $0 \le k \le 3$

- Code fragments used are

```
G = [4  1+7*i  2  1-7*i];
g = ifft(G);
```

which yield

```
g =
  2  -3  1  4
```

Relation Between the DTFT and DFT

- The DTFT $X(e^{j\omega})$ of a length-N sequence $x[n]$ is given by

$$X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n} = \sum_{n=0}^{N-1} x[n]e^{-j\omega n}$$

- We next evaluate samples of $X(e^{j\omega})$ at N equally spaced frequencies:

$$X(e^{j\omega_k}) = X(e^{j\omega})\Big|_{\omega=2\pi k/N},\ 0 \le k \le N-1$$

Relation Between the DTFT and DFT

resulting in

$$X(e^{j\omega_k}) = \underbrace{\sum_{n=0}^{N-1} x[n]e^{-j2\pi k/N}}_{X[k]},\ 0 \le k \le N-1$$

- Thus, samples of the N-point DFT of a length-N time-domain sequence are simply the frequency samples of its DTFT evaluated at N equally spaced frequencies:

Relation Between the DTFT and DFT

$$\omega_k = 2\pi k/N,\ 0 \le k \le N-1$$

in the range $0 \le \omega < 2\pi$

- Consequently, the DFT is a frequency-domain representation of a finite-length sequence
- The real integer variable k sometimes is referred to as the frequency index

---

Relation Between the DTFT and DFT

- The location of the frequency index on the normalized frequency range $0 \le \omega < 2\pi$ is called the bin location
- The normalized angular frequency ω_k associated with the bin location k is simply $2\pi k/N$ radians

Relation Between the DTFT and DFT

Example - For a 64-point DFT of a length-64 sequence, the normalized angular frequency of the bin location $k = 14$ is $\omega = 28\pi/64 = 7\pi/16$ radians

Time-Domain Operations on Finite-Length Sequences

- A major constraint on applying a time-domain operation described earlier on a finite length sequence defined for a specified range of time indices is that the generated sequence must also be defined for the same range of the time indices

Time-Domain Operations on Finite-Length Sequences

- The time-shifting operation

$$y[n] = x[n - N_o]$$

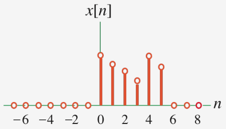
and the time-reversal operation

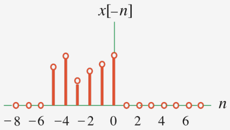
$$y[n] = x[-n]$$

are not applicable as the generated finite-length sequences are defined for different ranges of the time indices

Time-Domain Operations on Finite-Length Sequences

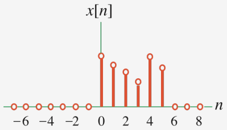
- For example, the conventional time-reversal of a length-N sequence $x[n]$ defined for $0 \le n \le N-1$ generates a length-N sequence $x[-n]$ that is defined for $-N+1 \le n \le 0$

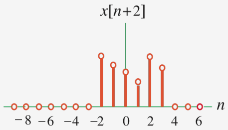




Time-Domain Operations on Finite-Length Sequences

- Likewise, the conventional time-shifting operation also generates a sequence which is outside the original range of the time indices as shown below





Time-Domain Operations on Finite-Length Sequences

- Similarly, the convolution sum operation when applied on two finite-length sequences of lengths N_1 and N_2 generates a finite-length sequence of length $N_2 + N_1 - 1$
- In order to be applicable to finite-length sequences defined for a specific range of the time indices, these operations need to redefined using the modulo operation

Circular Time-Reversal Operation

- The circular time-reversed version $y[n]$ of a length-N sequence $x[n]$ defined for $0 \le n \le N-1$ is given by

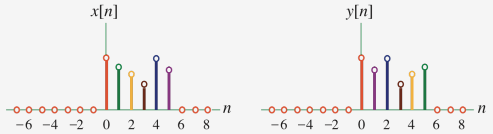
$$\boxed{y[n] = x[\langle -n \rangle_N],\ 0 \le n \le N-1}$$

where $\boxed{\langle -n \rangle_N = (-n) \ \text{modulo}\ N}$

Circular Time-Reversal Operation

- Figure shown below illustrates a length-6 sequence $x[n]$ and its circular time-reversed version

$$y[n] = x[\langle -n]\rangle_6 = x[6-n]$$



Circular Time-Reversal Operation

Example – Let $x[n] = \{\underset{\uparrow}{-3}, 5, 7, 0, -8, 9\}$

- We determine its circular time-reversed version $y[n]$:

$$y[0] = x[\langle -0 \rangle_6] = x[0] = -3$$
$$y[1] = x[\langle -1 \rangle_6] = x[6-1] = x[5] = 9$$
$$y[2] = x[\langle -2 \rangle_6] = x[6-2] = x[4] = -8$$
$$y[3] = x[\langle -3 \rangle_6] = x[6-3] = x[3] = 0$$
$$y[4] = x[\langle -4 \rangle_6] = x[6-4] = x[2] = 7$$
$$y[5] = x[\langle -5 \rangle_6] = x[6-5] = x[1] = 5$$

Circular Time-Shifting Operation

- Hence,

$$y[n] = \{-3, 9, -8, 0, 7, 5\}\ , 0 \le n \le 5$$

Circular Time-Shifting Operation

- For a finite length sequence $x[n]$ defined for $0 \le n \le N-1$, its circular time-shifted version $y[n]$, shifted by an integer amount M, is given in the next slide

Circular Time-Shifting Operation

$$\boxed{y[n] = x[\langle n-M \rangle_N],\ 0 \le n \le N-1}$$

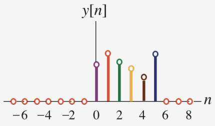
- If M is a positive integer, the above operation defines a right circular shift
- If M is a negative integer, the above operation defines a left circular shift
- For $M > 0$ with $1 \le M \le N-1$, we have

$$y[n] = \begin{cases} x[n-M], & \text{for } 1 \le M \le n \le N-1 \\ x[N+n\text{-}M], & \text{for } 1 \le n < M \le N-1 \end{cases}$$

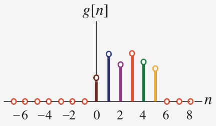
Circular Time-Shifting Operation

- It should be noted that if M is outside the range $0 < M \le N-1$, it is replaced by an integer $M_o = \langle M \rangle_N$
- The circular time-shifting operation is illustrated in the next slide

Circular Time-Shifting Operation



$\{y[n]\} = \{x[\langle n-7\rangle_6]\}$



$\{g[n]\} = \{x[\langle n+3\rangle_6]\}$

- Note: A left circular shift by M sample periods is equivalent to a right circular shift by $N-M$ sample periods, and vice-versa

---

Circular Time-Shifting Operation

Example – Let $x[n] = \{\underset{\uparrow}{-3},\ 5,\ 7,\ 0,\ -8,\ 9\}$

- We determine $y[n] = x[\langle n-7\rangle_6]$
- As $M = 7$ is greater than $N-1 = 6-1 = 5$, we replace it with $M_o = \langle 7\rangle_6 = 1$ and determine $y[n] = x[\langle n-1\rangle_6]$
- The 6 samples of $y[n]$ are given in the next slide

Circular Time-Shifting Operation

$y[0] = x[6+0-1] = x[5] = 9.3$

$y[1] = x[1-1] = x[0] = -3.1$

$y[2] = x[2-1] = x[1] = 5.5$

$y[3] = x[3-1] = x[2] = 4.7$

$y[4] = x[4-1] = x[3] = 0$

$y[5] = x[5-1] = x[4] = -8.2$

Circular Time-Shifting Operation

- We next determine $g[n] = x[\langle n+3\rangle_6]$
- As here $M = -3$, we replace it with $M_o = \langle -3\rangle_6 = 3$ and compute $g[n] = x[\langle n-3\rangle_6]$
- The 6 samples of $g[n]$ are given in the next slide
-

Circular Time-Shifting Operation

- The 6 samples of $g[n] = x[\langle n-3\rangle_6]$ are

$g[0] = x[6-3] = x[3] = 0$

$g[1] = x[6-3+1] = x[4] = -8$

$g[2] = x[6-3+2] = x[5] = 9$

$g[3] = x[6-3+3] = x[0] = -3$

$g[4] = x[4-3] = x[1] = 5$

$g[5] = x[5-3] = x[2] = 7$

Circular Convolution

- Recall that the linear convolution of two length-N sequences, $x[n]$ and $h[n]$, defined for $0 \le n \le N-1$, results in a length-$(2N-1)$ sequence $y_L[n]$ defined for $0 \le n \le 2N-2$:

$$y_L[n] = \sum_{\ell=0}^{N-1} x[\ell]h[n-\ell],\ 0 \le n \le 2N-2$$

Circular Convolution

- The circular convolution of two length-N sequences, $x[n]$ and $h[n]$, defined for $0 \le n \le N-1$, is given by

$$\boxed{y_C[n] = \sum_{\ell=0}^{N-1} x[\ell]h[\langle n-\ell\rangle_N],\ 0 \le n \le N-1}$$

- To indicate the size of the result, the above operation is usually called the N-point circular convolution

---

Circular Convolution

- The N-point circular convolution is often shown in compact form as

$$\boxed{y_C[n] = x[n] \textcircled{N} h[n]}$$

- The circular convolution operation is commutative and associative, that is,

$$\boxed{x[n] \textcircled{N} h[n] = h[n] \textcircled{N} x[n]}$$

$$\boxed{(x[n] \textcircled{N} h[n]) \textcircled{N} g[n] = h[n] \textcircled{N} (x[n] \textcircled{N} g[n])}$$

Circular Convolution

Example – Let

$$\{x[n]\} = \{\underset{\uparrow}{2},\ -3,\ 0,\ -4\},\{h[n]\} = \{\underset{\uparrow}{-2},\ 0,\ 5,\ -3\}$$

- The 4-point circular convolution $y_C[n]$ is given by

$$y_C[n] = \sum_{\ell=0}^{3} x[\ell]h[\langle n-\ell\rangle_4],\ 0 \le n \le 3$$

Circular Convolution

$$y_C[0] = \sum_{\ell=0}^{3} x[\ell]h[\langle 0-\ell\rangle_4]$$

$$= x[0]h[\langle 0\rangle_4] + x[1]h[\langle -1\rangle_4] + x[2]h[\langle -2\rangle_4] + x[3]h[\langle -3\rangle_4]$$

$$= x[0]h[0] + x[1]h[3] + x[2]h[2] + x[3]h[1]$$

$$=2\times(-2)+(-3)\times 0+0\times 5+9-4)\times(-3)=5$$

$$y_C[1] = \sum_{\ell=0}^{3} x[\ell]h[\langle 1-\ell\rangle_4]$$

$$= x[0]h[1] + x[1]h[0] + x[2]h[3] + x[3]h[2]$$

$$= 2\times 0 + (-3)\times(-2) + 0\times(-3) + (-4)\times 5 = -14$$

Circular Convolution

$$y_C[2] = \sum_{\ell=0}^{3} x[\ell]h[\langle 2-\ell\rangle_4] = 22$$

$$y_C[3] = \sum_{\ell=0}^{3} x[\ell]h[\langle 3-\ell\rangle_4] = -13$$

Hence,

$$y_C[n] = \{5,\ -14,\ 22,\ -13\},\ 0 \le n \le 3$$

DFT Properties

$$g[n] \overset{\text{DFT}}{\leftrightarrow} G[k] \qquad h[n] \overset{\text{DFT}}{\leftrightarrow} H[k]$$

Linearity Property:

$$\alpha g[n] + \beta h[n] \overset{\text{DFT}}{\leftrightarrow} \alpha G[k] + \beta H[k]$$

Circular Time-Shifting Property:

$$g[\langle n-n_o\rangle_N] \overset{\text{DFT}}{\leftrightarrow} W_N^{kn_o} G[k]$$

Circular Frequency-Shifting Property:

$$W_N^{-k_o n} g[n] \overset{\text{DFT}}{\leftrightarrow} G[\langle k-k_o\rangle_N]$$

DFT Properties

Duality Property:

$$G[n] \overset{\text{DFT}}{\leftrightarrow} Ng[\langle -k\rangle_N]$$

Circular Convolution Property:

$$\sum_{m=0}^{N-1} g[m]h[\langle n-m\rangle_N] \overset{\text{DFT}}{\leftrightarrow} G[k]H[k]$$

Multiplication Property:

$$g[n]h[n] \overset{\text{DFT}}{\leftrightarrow} \frac{1}{N}\sum_{m=0}^{N-1} G[m]H[\langle k-m\rangle_N]$$

Circular Convolution Using the DFT

Parseval's Relation

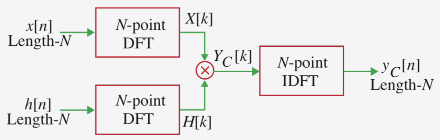
$$\sum_{n=0}^{N-1} |g[n]|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |G[k]|^2$$

- From the circular convolution property of the DFT we have

$$x[n] \textcircled{N} h[n] \overset{\text{DFT}}{\leftrightarrow} X[k]H[k]$$

Circular Convolution Using the DFT

- Hence, an alternate approach to determine the circular convolution of two length-N sequences is as indicated below



$x[n]$ Length-N

N-point DFT

$X[k]$

$Y_C[k]$

N-point IDFT

$y_C[n]$ Length-N

$h[n]$ Length-N

N-point DFT

$H[k]$

DFT Properties

Example – Let

$x[n] = \{\underset{\uparrow}{2},\ -3,\ 0,\ -4\}$, $h[n] = \{\underset{\uparrow}{-2},\ 0,\ 5,\ -3\}$

- Their 4-point DFTs are given by

$$X[k] = \{-5,\ 2-j,\ 9,\ 2+j\}$$
$$H[k] = \{0,\ -7-j3,\ 6,\ -7+j3\}$$

- The sample-wise products of the DFTs $X[k]$ and $H[k]$ are thus given by

---

DFT Properties

$$Y_C[k] = \{X[0]H[0],\ X[1]H[1],\ X[2]H[2],\ X[3]H[3]\}$$
$$= \{0,\ -17+j,\ 54,\ -17-j\}$$

- A 4-point IDFT of the above obtained using MATLAB is given by

$$y_C[n] = \{5,\ -14,\ 22,\ -13\},\ 0 \le n \le 3$$

Circular Convolution Using MATLAB

- Code fragments to compute the circular convolution of

  $x[n] = \{\underset{\uparrow}{2},\ -3,\ 0,\ -4\}$, $h[n] = \{\underset{\uparrow}{-2},\ 0,\ 5,\ -3\}$

  are given in the next slide

Circular Convolution Using MATLAB

```
x = [2 -3 0 -4];
h = [-2 0 5 -3];
X = fft(x);
H = fft(h);
Y = X.*H;
y = ifft(Y);
```

Linear Convolution Using the DFT

which results in

```
y =
   5 -14 22 -13
```

Linear Convolution Via the DFT

- Consider two finite length sequences, $x[n]$ and $h[n]$, of lengths N and M, respectively

---

Linear Convolution Using the DFT

- Their convolution sum is given by

$$y_L[n] = x[n] \circledast h[n]$$

which is of length $L = N + M - 1$

- To implement the convolution sum using the circular convolution we first extend the two sequences to length-L by appending them with zero-valued samples as indicated in the next slide

---

Linear Convolution Using the DFT

$$x_e[n] = \begin{cases} x[n], & 0 \le n \le N-1 \\ 0, & N \le n \le L-1 \end{cases}$$

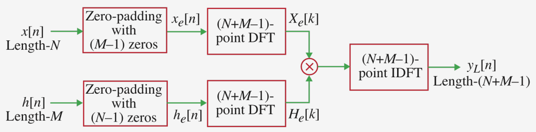
$$h_e[n] = \begin{cases} h[n], & 0 \le n \le M-1 \\ 0, & M \le n \le L-1 \end{cases}$$

- The linear convolution of $x[n]$ and $h[n]$ is then obtained by computing

$$y_L[n] = x[n] \circledast h[n] = \underbrace{x_e[n] \textcircled{L} h_e[n]}_{L\text{-point circular convolution}}$$

Linear Convolution Using the DFT

- The proposed approach is shown below



Linear Convolution Using the DFT

Example – We develop the linear convolution of $x[n] = \{2,\ -3,\ 0,\ -4\}$, $0 \le n \le 3$ and $h[n] = \{-2,\ 0,\ 5,\ -3\}$, $0 \le n \le 3$ using the DFT-based approach in MATLAB

- Code fragments used are shown in the next slide

Linear Convolution Using the DFT

```
x = [2 -3 0 -4];
h = [-2 0 5 -3];
XE = fft(x,7);
HE = fft)h,7);
YL = XE.*HE;
yL = ifft(YL);
```

Linear Convolution Using the DFT

which yields

```
yL =
Columns 1 through 7
-4.0   6.0   10.0   -13.0   9.0
-20.0  12.0
```

Linear Convolution Using the DFT

- To verify the above result we compute the linear convolution of the original length-4 sequences using MATLAB
- Code fragments used are

```
x = [2 -3 0 -4];
h = [-2 0 5 -3];
yL = conv(x,h)
```

Linear Convolution Using the DFT

which yields

```
yL =
-4.0   6.0   10.0   -13.0   9.0
-20.0  12.0
```

as expected